A Parallel Algorithm for Computing Shortest Paths in Large-Scale Networks^{*}

Guozhen Tan and Xiaohui Ping

Department of Computer Science, Dalian University of Technology,
Dalian, 116023, P.R. China
gztan@dlut.edu.cn

Abstract. This paper presents the Optimality Theorem in distributed parallel environment. Based on this theorem, a parallel algorithm using network-tree model is presented to compute shortest paths in large-scale networks. The correctness of this algorithm is proved theoretically and a series of computational test problems are performed on PC cluster. Factors such as network size and level of the network, which take effect on the performance, are discussed in detail. Results of the experiments show that the proposed parallel algorithm is efficient in computing shortest paths in large-scale networks, especially when the network size is great.

1 Introduction

The recent research interest in large-scale networks has been focused on partitioning the network and constructing new network model. The previous contributions are the Level graph model[2][3], the Hierarchical Encoded Path View (HEPV) model[4] and the Network-Tree Model (NTM)[5]. These models all have some defects. The Network-Tree Model can be constructed from arbitrary network, and the route optimization can be constrained within a few sub-networks, which greatly reduce the searching scope and significantly improve the computing efficiency.

In this paper, we present a parallel algorithm based on network-tree model to solve the problem and test it in a series of networks. The experiment results validate that the parallel algorithm achieve high performance in speedup and efficiency.

2 Network-Tree Model

Network-Tree Model is the definitional combination of network and tree. Any network can be divided into a number of sub-networks, and each sub-network will be constructed to be one node of network-tree. We use macro-node to denote the node of network-tree to distinguish from the node of original network.

^{*} This work is supported by the National Science Foundation of China(60373094).

Network-Tree Model(NTM): In a given network N , let the network-tree be $N_Tree = (T, H)$, where T is the macro-node set of N . If only one macro-node exists in T , $H = \phi$, otherwise, H is a binary relation defined on T . There is only one macro-node m_r called the root of N_Tree . If $T - \{m_r\} \neq \phi$, there exists a collection of disjoint trees $T_1, T_2, \dots, T_n (n > 0)$. For arbitrary $i (1 \leq i \leq n)$, there is one unique macro-node $m_i \in T_i$ satisfies $\langle m_r, m_i \rangle \in H$. For $H - \{\langle m_r, m_1 \rangle, \dots, \langle m_r, m_n \rangle\}$, there exists a collection of disjoint binary relations $H_1, H_2, \dots, H_n$, and $\forall i, 1 \leq i \leq n$, H_i is a binary relation defined on T_i , and (T_i, H_i) is also a network-tree defined above, and we call it the sub-tree of the root macro-node m_r .

Definition 1: In network-tree model $N_Tree = (T, H)$, for any macro-node $m_c \in T$ and its parent macro-node $m_p \in T$, the node set of $m_c \cap m_p$ is called the connecting node set of m_c , denoted by R_{m_c} .

Let $sp_m(u, v)$ denote the shortest path from u to v in network m , $sp_{T'}(u, v)$ denote the shortest path from u to v in network-tree T' , and $l(\pi)$ denote the distance of path π .

Definition 2: Network-tree model $N_Tree = (T, H)$, if for each sub-tree T' of N_Tree , with macro-node m_r being the root of T' , $\forall u, v \in V(m_r)$, $l(sp_{m_r}(u, v)) = l(sp_{T'}(u, v))$, thus N_Tree is called Expanded Network-Tree Model.

Definition 3: Network-tree model $N_Tree = (T, H)$, let the transitive closure of macro-node m be $B_m = \{m_p | (m_p \in T) \wedge (m_p = m \vee (\exists m_q \in B_m) (\langle m_p, m_q \rangle \in H))\}$.

Definition 4: In network model $N = (V, A, W)$, $\forall u, v \in V$, let the virtual arc from u to v be $\overline{a_{uv}} = (u, v, sp_N(u, v))$, with $w(\overline{a_{uv}}) = l(sp_N(u, v))$.

3 A Parallel Algorithm Based on Network-Tree Model

3.1 Optimality Theorem

Theorem 1: In the Expanded Network-Tree Model, for given nodes s and t , the optimal path π from s to t can be represented as following equation: $\pi = sp_{m_1}(s = v_0, v_1) + sp_{m_2}(v_1, v_2) + \dots + sp_{m_h}(v_{h-1}, v_h) + sp_{m_{h+1}}(v_h, v_{h+1}) + sp_{m_{h+2}}(v_{h+1}, v_{h+2}) + \dots + sp_{m_{n-1}}(v_{n-2}, v_{n-1}) + sp_{m_n}(v_{n-1}, v_n = t)$ where $n \geq 1, 1 \leq h \leq n, \langle m_h, m_{h-1} \rangle \in H, \langle m_{h-1}, m_{h-2} \rangle \in H, \dots, \langle m_2, m_1 \rangle \in H$, and $\langle m_{h+1}, m_{h+2} \rangle \in H, \dots, \langle m_{n-1}, m_n \rangle \in H, v_{i+1} \subseteq R_{m_{i+1}}$ with $0 \leq i < h$, and $v_i \subseteq R_{m_{i+1}}$ with $h < i < n$.

3.2 Parallel Dividing Method and Distributed and Parallel Network Tree Shortest Path Algorithm (DP_NTSP Algorithm)

First we get the expanded network-tree through preprocessing. According to theorem 1, the optimal path is consisted of a series of sub-paths in each macro-node of the transitive closures. Then we pre-compute the distances from arbitrary

node to connecting nodes in each macro-node, and write the distance data to a file, so the master processor can compute an optimal shortest path consisting of virtual arcs according to the distances. Then the master distributes the virtual arcs to the slaves, and each slave computes the required actual path.

Description of *DP_NTSP* Algorithm. We use the multiple-program-multiple-data (MPMD) model to implement the parallel algorithm. Suppose the source node is s , and the destination node is t .

Master program:
Step1 : Creates p copies of the slave program.
Step2 : According to s and t , computes the transitive closures $B_{\sigma(s)}$ and $B_{\sigma(t)}$, let $B = B_{\sigma(s)} \cup B_{\sigma(t)}$.
Step3 : Reads the distance data contained by B from the file and computes an optimal path consisting of virtual arcs.
Step4 : While there is an active slave program,
 4[a] receives the results from the slave program.
 4[b] If all the virtual arcs have not been allocated, sends a new virtual arc to the slave program i ; Otherwise, stop the execution of slave i .
Step5 : Creates a shortest path according to the received results and terminates.

Slave program:
Step1 : Creates the code for the computation of shortest paths.
Step2 : While there is a virtual arc to handle,
 2[a] receives from the master program a virtual arc or the instruction to stop execution.
 2[b] gets the network data that related to the virtual arc and computes the actual path that corresponds to the virtual arc.
 2[c] sends the results to the master program.

4 Computational Results

In our computational work, we choose six networks. The detailed parameters of each network are shown in table 1. We test the execution time of *DP_NTSP* algorithm in each network; the computational results are presented in Fig.1.

Table 1. parameters of the testing networks

Network	k	$l_1 = b_1$	$l_2 = b_2$	$l_3 = b_3$	w_1	w_2	w_3	Nodes	Arcs
g_1	2	6	10	/	5	9	/	3,721	14,640
g_2	2	8	8	/	9	5	/	4,225	16,640
g_3	3	2	5	8	6	6	6	6,561	25,920
g_4	3	4	4	6	2	3	6	9,409	37,248
g_5	3	6	6	6	5	6	8	47,089	187,488
g_6	3	6	7	8	5	7	9	113,569	452,928

Fig. 1. Calculation time in different networks

Fig. 2. speedup curves

Fig. 3. efficiency curves

From Fig.1 we can see that with the number of hosts increasing, the execution time of algorithm is relatively decreased. In Fig.2 and Fig.3, it is apparent that with the network scale increasing, the speedup and efficiency of the algorithm are improved notably. The results show that the proposed parallel algorithm can well solve the optimal path problem in large-scale networks.

References

1. Koutsopoulos. Commentaries on Fastest Paths in Time-dependent Networks for IVHS Application. IVHS Journal, 1993, 1(1): 89-90
2. J.Shapiro, J.Waxman D.Nir. Level Graphs and Approximate Shortest Path Algorithms, Networks, 1992, 22: 691-717
3. Chan-Kyoo Park, Kiseok Sung, Seungyong Doh, Soondal Park. Finding a path in the hierarchical road networks. 2001 IEEE Intelligent Transportation Systems Conference 1998, 10 (2): 163-179
4. Ning Jing, Yun-Wu Wang, and Elke A. Rundensteiner. Hierarchical Encoded Path Views for Path Query Processing: An Optimal Model and Its Performance Evaluation. IEEE Transactions on Knowledge and Data Engineering, 1998, 10(3): 409 432
5. Guozhen Tan, Xiaojun Han, and Wen Gao. Network-Tree Model and Shortest Path Algorithm. The 3rd International Conference on Computational Science. Melbourne, Australia: Springer, Lecture Notes on Computer Science (LNCS 2659), 2003: 537-547